MORRISON & FOERSTER LLP
MICHAEL A. JACOBS (Bar No. 111664)
mjacobs@mofo.com
KENNETH A. KUWAYTI (Bar No. 145384)
kkuwayti@mofo.com
MARC DAVID PETERS (Bar No. 211725)
mdpeters@mofo.com
DANIEL P. MUINO (Bar No. 209624)
dmuino@mofo.com
755 Page Mill Road, Palo Alto, CA 94304-1018
Telephone: (650) 813-5600 / Facsimile: (650) 494-0792

BOIES, SCHILLER & FLEXNER LLP
DAVID BOIES (Admitted *Pro Hac Vice*)
dboies@bsfllp.com
333 Main Street, Armonk, NY 10504
Telephone: (914) 749-8200 / Facsimile: (914) 749-8300
STEVEN C. HOLTZMAN (Bar No. 144177)
sholtzman@bsfllp.com
1999 Harrison St., Suite 900, Oakland, CA 94612
Telephone: (510) 874-1000 / Facsimile: (510) 874-1460

ORACLE CORPORATION
DORIAN DALEY (Bar No. 129049)
dorian.daley@oracle.com
DEBORAH K. MILLER (Bar No. 95527)
deborah.miller@oracle.com
MATTHEW M. SARBORARIA (Bar No. 211600)
matthew.sarboraria@oracle.com
500 Oracle Parkway, Redwood City, CA 94065
Telephone: (650) 506-5200 / Facsimile: (650) 506-7114

Attorneys for Plaintiff
ORACLE AMERICA, INC.

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN FRANCISCO DIVISION

ORACLE AMERICA, INC.

Plaintiff,

v.

GOOGLE INC.

Defendant.

Case No. CV 10-03561 WHA

ORACLE'S APRIL 3, 2012 BRIEF REGARDING COPYRIGHT ISSUES

Dept.: Courtroom 8, 19th Floor
Judge: Honorable William H. Alsup

TABLE OF CONTENTS

TABLE OF CONTENTS
(continued)

Page

TABLE OF AUTHORITIES

Page(s)

CASES

STATUTES

OTHER AUTHORITIES

Pursuant to the Court's recent orders (ECF Nos. 835, 837, 838, and 843), Oracle responds to the Court's questions and addresses issues discussed at the March 28 hearing.

I. ORACLE'S ANSWERS TO THE COURT'S QUESTIONS

A. To what extent does Oracle contend that the 37 APIs as a group constitute a "compilation" within the meaning of Section 101, and/or contend that the individual APIs individually constitute a "compilation." Is there any other theory of protectability beyond "compilation" for the APIs?

The 37 APIs should not be viewed as a compilation under section 101 of the Copyright Act. Section 101 defines "compilation" as "a work formed by the collection and assembling of preexisting materials or of data that are selected, coordinated, or arranged in such a way that the resulting work as a whole constitutes an original work of authorship." The APIs are original and not a collection of preexisting materials or data. In *American Dental Ass'n. v. Delta Dental Plans Ass'n.*, 126 F.3d 977, 980 (7th Cir. 1997), for example, the Court held the ADA code of dental procedures is not a compilation:

> Note that we do *not* conclude that the Code is a compilation covered by 17 U.S.C. § 103. It could be a compilation only if its elements existed independently and the ADA merely put them in order. A taxonomy is a way of *describing* items in a body of knowledge or practice; it is not a collection or compilation *of* bits and pieces of "reality."

Id. at 980.

The 37 APIs are original works of authorship. The specifications, as written documentation, and the class libraries (source or object code implementations of those specifications), as computer programs, are both protected as literary works. *See* 17 U.S.C. §§ 101, 102(a)(1); *Recording Indus. Ass'n of Am. v. Diamond Multimedia Sys., Inc.*, 180 F.3d 1072, 1077 (9th Cir. 1999) ("a computer program is not a material object, but rather, a literary work."). And even if individual API elements were not protectable, the combination of these elements would be. In *Satava v. Lowry*, for example, the Ninth Circuit found the individual elements of the plaintiff's jellyfish sculptures were unprotectable *scenes a faire*, but nonetheless went on to assess whether the combination was copyrightable. 323 F.3d 805, 811 (9th Cir. 2003). The court stated:

> Our case law suggests, and we hold today, that ***a combination of unprotectable elements is eligible for copyright protection only if those elements are numerous enough and their selection and arrangement original enough that their***

> ***combination constitutes an original work of authorship.*** *See Metcalf*, 294 F.3d at 1074; *Apple Computer, Inc.*, 35 F.3d at 1446. *See also Feist*, 499 U.S. at 358, 111 S.Ct. 1282 ("[T]he principal focus should be on whether the selection, coordination, and arrangement are sufficiently original to merit protection.").

Id. (emphasis added).

Google's recent admission gives it little room to argue. Google concedes originality:

> Google does not dispute that the APIs as a whole meet the "extremely low" threshold for originality required by the Constitution. *Feist Pubs., Inc. v. Rural Tele. Serv. Co.*, 499 U.S. 340, 345 (1991). The jury therefore need not be asked to address whether the APIs are original.

(ECF No. 823 at 9). There can be no question of "numerosity." The 37 API packages encompass thousands of elements and the many relationships among them. Both of the elements under *Satava* are therefore satisfied. Google's admission establishes copyrightability.

B. (1) Under the law, does "selection, arrangement, and structure" arise as an issue only in the context of originality and more specifically, as a way to allow protectability for otherwise noncopyrightable elements in a compilation?

No. It is true that, under section 101 of the Copyright Act, to qualify for copyright protection a compilation must be "selected, coordinated, *or* arranged in such a way that the resulting work as a whole constitutes an original work of authorship." 17 U.S.C. § 101 (emphasis added). Satisfying any one of these requirements is sufficient. However, these concepts are not limited to compilations or to "otherwise noncopyrightable elements" and they do not arise "only in the context of originality." Selection, arrangement, and structure collectively and individually also relate to expression, not just originality. A unique and detailed structure, like that of the 37 APIs, qualifies as expression. *See, e.g., Johnson Controls, Inc. v. Phoenix Control Sys., Inc.*, 886 F.2d 1173, 1176 (9th Cir. 1989) (finding structure of computer program was "expression").

C. (2) Is "selection, arrangement, and structure" also an exception to the rule that a system or method cannot be copyrighted? That is, if something is a system or method can it still be copyrighted so long as the system or method is the result of selection, arrangement and structure?

No. To the extent a system or method of operation is an idea, it cannot be copyrighted. But the decisional law establishes that proper copyright analysis must move beyond labels. Original expression in a work can be copyrighted, even if the work also describes or embodies an

idea. That includes selection, arrangement or structure. *See id.*at 1175. ("Whether the non-literal components of a program, including the structure, sequence and organization and user interface, are protected depends on whether, on the particular facts of each case, the component in question qualifies as an expression of an idea, or an idea itself.")

The rule that the Court refers to here is the idea/expression dichotomy codified in Section 102(b), which is discussed in section II.B below. Oracle's selection, arrangement, and structure of API elements as fixed in the API specifications and class libraries are original expression, and are protected by copyright. The underlying ideas are not.

D. (3) If a method or system is copyrightable if the result of selection, arrangement, and structure, won't that be true for all original methods and systems which by definition involve a structure, arrangement, and selection steps?

No. The idea of a system or method of operation is not protectable. A work that expresses or embodies a system or method will be copyrightable to the extent it contains original expression, which may or may not relate to selection, arrangement, or structure. Thus the premise of this question is not correct. Nor is it quite correct that all methods and systems "by definition" involve structure, arrangement, and selection steps.

The copyrightability of works that embody systems depends on the facts. Some works do not pass muster, but not necessarily because they are "systems." In *Toro Co. v. R & R Prods. Co.*, for example, the court found the plaintiff's parts numbering system was unprotectible because it was unoriginal, not because it was a system under § 102(b). 787 F.2d 1208, 1212 (8th Cir. 1986) ("the district court's literal application of the section's language - that appellant's parts numbering system is not copyrightable because it is a 'system' - cannot stand.").

E. (4) For the merger doctrine, at what level of abstraction should we consider the idea/system? At a high level of abstraction (for instance, the concept of APIs generally), there are many ways to express "selection, arrangement, and structure" in creating a particular API. But for the idea/system of the 37 Java APIs, there may be only one way to express the "selection, arrangement, and structure."

Oracle is not claiming copyright protection for the idea of APIs generally, or for the idea of creating APIs that provide particular capabilities, *e.g.*, input/output or security. It claims protection for the particular expression contained in the 37 API packages: the many elements it

chose to include in its design, the arrangement of these elements, and all of the relationships among them. While it can sometimes be difficult to separate idea from expression, that is not the case here, where Google copied 11,000 pages of API specifications. No case has held that a structure this detailed and complex is an "idea" rather than the expression of an idea.

As has been discussed in more detail elsewhere, even at the package level, the authors of the APIs made choices that could have been very different. The packages java.util and java.nio reflect different design choices and illustrate the originality and creativity of the Java APIs. When Sun chose to create an improved input/output API ("New I/O") to supplement the existing java.io package, it chose to arrange the new related input/output elements into six different API packages instead of one. *See* http://en.wikipedia.org/wiki/Java.nio. (*See* ECF No. 611 at 2-3.) In contrast, the java.util package specification is a single "grab bag" of 49 classes and 16 interfaces that perform unrelated functions and did not all have to be placed in one package. The APIs are replete with design choices, large and small, that reflect the expressiveness of their designers.

The Ninth Circuit articulated how the line between idea and expression should be drawn in protecting the collectable coin pricing guide in *CDN Inc. v. Kapes*:

> As Judge Hand noted, the difference between idea and expression is one of degree. This circuit has held that "***[t]he guiding consideration in drawing the line is the preservation of the balance between competition and protection reflected in the patent and copyright laws."*** *Rosenthal*, 446 F.2d at 742. In this case, the prices fall on the expression side of the line. ***CDN does not, nor could it, claim protection for its idea of creating a wholesale price guide, but it can use the copyright laws to protect its idea of what those prices are***. *See id.* at 742 (denying protection to the idea of creating a jeweled bee pin where there was no indication that the alleged infringer had copied the pin in question). Drawing this line preserves the balance between competition and protection: it allows CDN's competitors to create their own price guides and thus furthers competition, but protects CDN's creation, thus giving it an incentive to create such a guide.

197 F.3d 1256, 1262 (9th Cir. 1999) (emphasis added). The 37 APIs are far more expressive than *CDN*'s coin pricing guide, but the same principle applies. The 37 APIs have been developed and refined over the course of many years. While Oracle cannot claim protection for the idea of an API or a type of API package, it can for the detailed structure it created. This preserves the incentive for companies to invest in developing APIs, but allows competitors to create their own. Google has proven that it could when it wishes to—Android includes more than 100 new APIs.

F. (5) Is it true that the APIs are an integral part of Java? That is, programmers write their own programs using the APIs? If the answer depends on the particular API, please specify which of the 37 asserted are integral.

The 37 asserted APIs are an integral part of the Java Platform, but are not an integral part of the Java programming language. The Java programming language requires only a very small number of API elements, as Oracle has discussed previously. (ECF No. 780 at 15.)As Google alleged in its pleadings, the Java programming language is ***distinct*** from the Java Platform, which includes the APIs (the class libraries): "***While they are distinct elements***, the term "Java" is commonly used to refer to the programming language, the runtime environment, as well as the platform." (Google Amended Counterclaims ¶ 1, ECF No. 51 at 13 (emphasis added).) Google defines a Java runtime environment as "consisting of Java virtual machines written to operate on a number of different computer platforms and ***a set of standard class libraries*** that can be accessed and reused by Java platform applications to perform common software functions, such as writing to files or sorting data." (*Id.* ¶ 3 at 14 (emphasis added).).

Programmers do write their own programs using the APIs. What this means is that application programmers typically write only a portion of their programs themselves, and choose to call upon the pre-written class libraries described by Oracle's APIs in their application to provide the remainder. An application written this way will be incomplete: it will not itself contain all of the code needed to carry out the functions of the application. Only someone who also has an executable copy of the API-associated class libraries that the application can call upon can run such an application. Thus, while an application programmer does not need a license to the APIs from Oracle to author and distribute a program in the Java programming language (even if it includes calls to the APIs), whoever runs a program that includes API calls will need a license from Oracle, because that person needs an executable implementation of the APIs.

In practice, the way most programmers and users obtain an executable implementation of the APIs is by downloading it from Oracle in the form of the Java Development Kit (for programmers) and the Java Runtime Environment (for users). Both the JDK and JRE are subject to a "non-exclusive, non-transferable, limited license without license fees" that permits use and

redistribution for desktop and server computers ***only***, and that prohibits modification and reverse engineering. *See* http://www.oracle.com/technetwork/java/javase/terms/license/index.html. These aspects of Oracle's licensing model are the main reason people think "Java" is "free."

As discussed in section IV below, Oracle also licenses its API specifications for clean-room, complete, and fully compatible reimplementations that pass the TCK test suite (which has its own license that must be separately accepted). Oracle does ***not*** license the Java APIs for mobile telephones and other embedded applications except under the royalty-free GPL or a royalty-bearing commercial license.

G. (6) To what extent are computer languages (not programs, but languages) copyrightable? Patentable?

Oracle is not aware of any federal judicial decision holding that an artificial language is not copyrightable. Code books are copyrightable, and can be considered an artificial vocabulary:

> These words have a prospective meaning, but as yet they have not received, it, like an empty pitcher. Suppose some one devised a set of words or symbols to form a new abstract speech, with inflections, but as yet with no meaning, a kind of blank Esperanto. The case would be approaching the plaintiff's, though not there, because the words would, indeed, express relationship.

Reiss v. Nat'l Quotation Bureau, Inc., 276 F. 717, 718 (S.D.N.Y. 1921) (Hand, J.); *see also Hartfield v. Peterson*, 91 F.2d 998, 999 (2d Cir. 1937) ("Both the phrases, so far as they were his, and the arrangement were proper subjects of copyright."). A computer language should not be treated differently.

Nor is Oracle aware of any federal judicial decision holding that an artificial language is not patentable. Morse Code was held to be patentable in *O'Reilly v. Morse*, 56 U.S. (15 How.) 62, 86, 112 (1853) (upholding claim 5 of the 1948 reissue patent, which claims the "system of signs"). But Morse Code may not be a language, as it lacks a vocabulary and grammar.

H. (7) From Google's presentation at the hearing, it seems as though Sun/Oracle attempted to claim the structure/hierarchy/arrangement of APIs in the '855 patent and '093 patent. Would it be possible to claim the selection of classes for APIs under patent law?

It may in some case be possible to claim a selection of classes, so long as it was part of a "new and useful process, machine, manufacture, or composition of matter" under 35 U.S.C. § 101. Computer software is eligible for both patent and copyright protection. The Federal Circuit,

applying Ninth Circuit copyright law, so held in *Atari Games Corp. v. Nintendo of Am., Inc.*, 975 F.2d 832, 839-40 (Fed Cir. 1992) ("Title 35 protects the process or method performed by a computer program; title 17 protects the expression of that process or method. While title 35 protects any novel, nonobvious, and useful process, title 17 can protect a multitude of expressions that implement that process.").

The '855 and '093 patents reinforce the distinction, however, between patented processes or methods and the many different ways to express those ideas in API designs through the selection and design of classes. The '855 and '093 patents are software inventions that perform particular functions: The '855 patent is directed to methods and systems for integrating an application with a collaboration server, and the '093 patent is directed to class structures associated with clients and servers that enable access to a system database. *See* U.S. Pat. No. 7,996,855, claims 1 and 11; U.S. Pat. No 6,598,093, claims 1, 8 and 11.

These patents illustrate that a particular API may contain both expressive and functional content at the same time, and therefore benefit from both copyright and patent protection. What the patents describe and claim is more general than any specific API, leaving ample room for expressive choices. For example, the '093 patent states that the invention does not require any particular selection of classes or hierarchical relationships:

> The class hierarchy associated with a core API may generally vary. For instance, the class hierarchy may vary for embodiments in which there are either fewer or more classes associated with APIs. Similarly, the classes, as well as the methods, associated with different APIs may vary depending upon the requirements of a particular system without departing from the spirit or the scope of the present invention.

'093 Patent at 11:43-49. There are countless ways to design an API, through the selection, arrangement, and structure of API elements, to perform the patented method.

Notably, the '855 patent includes a copyright notice per 37 C.F.R. § 1.71(e), putting the public on notice that a portion of the patent document is subject to copyright protection. '855 Patent at 1:6-14. This is not unusual: Google likewise has issued patents—some including source code as part of the disclosure—over which it also claims copyright protection. *See, e.g.*, U.S. Pat. Nos. 8,111,690 and 7,953,152.

I. (8) By claiming that Google infringes the API implementation, is Oracle alleging that Google copied something other than the "selection, arrangement, and structure" of APIs, as fixed in the specifications?

Yes. The selection, arrangement, and structure of the Java APIs is not just "fixed in the specifications" but is also fixed in the source and object code implementation of the APIs in the Java class libraries. Similarly, Google copied the Java APIs into Android documentation and source code.

Oracle is also claiming copyright protection for the thousands of elements Google copied from the Java APIs into Android source code and documentation, and for the selection and arrangement of the names of the API elements that Google copied. And it is claiming that Google created derivative works from the English-language descriptions of the elements that are set forth in the API specifications.

J. (9) Discussion of CONTU

See section II.B. below.

K. (10) Critique of Prof. Samuelson's Article and Identification of Other Articles

Prof. Samuelson advocates an interpretation of § 102(b) that would reduce or eliminate copyright protection for functional works, however expressive or original. Her argument is that § 102(b) is not limited to a restatement of the idea/expression dichotomy, but rather that "all eight words of exclusion were put into the statute for a sound reason and that those who read the other seven words out of the statute are mistaken." Samuelson, *Why Copyright Law Excludes Systems and Processes from the Scope of Its Protection*, 85 Tex. L. Rev. 1921, 1923 (2007).

This view is contrary to the law. Nearly every Circuit has held that Section 102(b) is a restatement of the idea/expression dichotomy.[1] Prof. Jane Ginsburg has criticized the article's

[1] *See, e.g.*, *CCC Info. Servs., Inc. v. Maclean Hunter Mkt. Reports, Inc.*, 44 F.3d 61, 69 & n.12 (2d Cir. 1994); *Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222, 1234 (3d Cir. 1986); *M. Kramer Mfg. Co. v. Andrews*, 783 F.2d 421, 434 (4th Cir. 1986); *Kepner-Tregoe, Inc. v. Leadership Software, Inc.*, 12 F.3d 527, 533 & n.8 (5th Cir. 1994); *Atari, Inc. v. N. Am. Philips Consumer Elecs. Corp.*, 672 F.2d 607, 615 (7th Cir. 1982); *Toro*, 787 F.2d at 1211-12; *Apple Computer, Inc. v. Microsoft Corp.*, 35 F.3d 1435, 1443, n.11 (9th Cir. 1994); *Autoskill Inc. v. Nat. Educ. Support Sys., Inc.*, 994 F.2d 1476, 1491 (10th Cir. 1993); *Cable/Home Commc'n Corp. v. Network Prods., Inc.*, 902 F.2d 829, 842-43, n.26 (11th Cir. 1990).

approach and concluded § 102(b) cannot be interpreted literally as the article does, because otherwise the definition of a computer program in § 101 would be a process or method under §102(b). Ginsberg, *Four Reasons And A Paradox*, 94 Colum. L. Rev. 2559, 2569-70 (1994).

The article relies on the cases that support its position, overlooking others. It argues that "*Baker* sought to sharpen the distinction between patents and copyrights" (85 Tex. L. Rev. at 1933) and quotes the Seventh Circuit to the effect that Congress provided "two separate and distinct fields of protection, the copyright and the patent . . . . While it may be difficult to determine in which field protection must be sought, it cannot be found in both." *Id.* at 1937 (quoting *Taylor Instrument Cos. v. Fawley-Brost Co.*, 139 F.2d 98, 99 (7th Cir. 1943)). But the Article gives short shrift to the directly contrary Supreme Court holding in *Mazer v. Stein*: "Neither the Copyright Statute nor any other says that because a thing is patentable it may not be copyrighted." *Id.* at 1958 (quoting *Mazer*, 347 U.S. 201, 217 (1954)). The Article observes in a footnote that cases that interpret *Baker* and 102(b) to find original medical coding systems copyrightable, including the Ninth Circuit's decision in *Practice Mgmt. Info. Corp. v. Am. Med. Ass'n*, 121 F.3d 516, 518-19 (9th Cir. 1997), would reach a different result under the article's interpretation of *Baker* and 102(b). *Id.* at 1975 n. 350. The article presents a policy point of view. It does not purport to interpret the law.

Oracle agrees with the article in one respect: "The First Circuit's discussion of § 102(b) as applied to the Lotus command structure was not particularly well developed or compelling." *Id.* at 1971. As discussed below in section II.B, under Google's interpretation of *Lotus*, nearly all computer programs would be uncopyrightable, which cannot be correct.

For a thorough analysis of *Lotus*, *Baker*, the relationship between Sections 101 and 102, and the central role of originality under *Feist*, Oracle recommends an article by the treatise author William Patry, currently Google's Chief Copyright Counsel. William F. Patry, *Copyright and Computer Programs*, 14 Cardozo Arts & Ent. L.J. 1 (1996). Mr. Patry discusses the erroneous factual and legal premises on which the *Lotus* court based its decision. *Id.* at 13. He also provides a useful analysis of *Baker*, *id.* at 45-49, and an overview of the history of copyright protection of computer programs. *Id.* at 22-32. For the views of one CONTU Commissioner, as

well as a specific discussion of the case against any diminished copyright status for interfaces, Oracle recommends Arthur R. Miller, *Copyright Protection For Computer Programs, Databases, And Computer-Generated Works: Is Anything New Since CONTU?*, 106 Harv. L. Rev. 977, 1034 (1993) ("Any limitation on the protection of 'interfaces' thus would be a limitation on the protection of much of the valuable expression in programs, and would invite plagiarists to label as an 'interface' whatever they have chosen to copy without permission.").

II. *JOHNSON CONTROLS* IS NINTH CIRCUIT CONTROLLING AUTHORITY

A. The Copyrightability of Non-Literal Components of Computer Programs Is Governed By *Johnson Controls*

In *Johnson Controls*, the Ninth Circuit held that non-literal components of a computer program are copyrightable if they qualify as an expression of an idea, rather than the idea itself. 886 F.2d at 1175. At issue was the copyrightability of the "structure, sequence and organization" of the plaintiff's JC-5000S computer program. *Id.* The Ninth Circuit held that:

> Whether the non-literal components of a program, including the structure, sequence and organization and user interface, are protected depends on whether, on the particular facts of each case, the component in question qualifies as an expression of an idea, or an idea itself.

Id. Pointing to evidence that "some discretion and opportunity for creativity exist in the structure," the Ninth Circuit upheld the lower court's finding, on preliminary injunction, that "the structure of the JC-5000S is expression, rather than an idea in itself." *Id.* at 1176.

Johnson Controls remains the law of this Circuit. Citing *Johnson Controls*, the Federal Circuit held that the computer program key to Nintendo's video console was protected by copyright, stating that "[t]his court, in applying Ninth Circuit law, must determine whether each component of the 10NES program 'qualifies as an expression of an idea, or an idea itself.'" *Atari v. Nintendo*, 975 F.2d at 840 (quoting *Johnson Controls*, 886 F.2d at 1175). *See also Merch. Transaction Sys., Inc. v. Nelcela, Inc.*, 2009 U.S. Dist. LEXIS 25663, at *38 (D. Ariz. Mar. 17, 2009) (following *Johnson Controls* and finding that "the selection, coordination, and arrangement of the information contained in the Lexcel software's database schema" constituted copyrightable subject matter).

B. The *Lotus* Approach Is Incorrect And Does Not Apply Here

Google relies on *Lotus* to argue that expressive choices do not matter once a court determines that a work is a method of operation: "Accepting the district court's finding that the Lotus developers made some expressive choices in choosing and arranging the Lotus command terms, we nonetheless hold that that expression is not copyrightable because it is part of Lotus 1-2-3's 'method of operation.'" *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 49 F.3d 807, 816 (1st Cir. 1995), *aff'd by an evenly divided court*, 516 U.S. 233 (1996).

Lotus defined "methods of operation" as "the means by which a user operates something." *Id.* at 815. This definition is dangerously close to the statutory definition of "computer program": "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." 17 U.S.C. § 101. Congress adopted this definition from the Final Report of the National Commission on New Technological Uses of Copyrighted Works (1979) ("CONTU Report"). CONTU's first recommendation was "to make it explicit that computer programs, to the extent that they embody an author's original creation, ***are proper subject matter of copyright***." CONTU Report at 1 (emphasis added). "Congress adopted all of the statutory changes recommended by CONTU verbatim. Subsequent Congresses, the courts, and commentators have regarded the CONTU Report as the authoritative guide to congressional intent." *Sega Enters. Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1520 n.5 (9th Cir. 1993). All computer programs are a means to bring about a certain result. Because most programs could be considered a means of operating something, *Lotus* is inconsistent with Congress's intent to protect software thought copyright.

That a developer might make use of a class library associated with an API—or any other software—does not diminish that software's copyrightability. The Ninth Circuit relied on the CONTU Report in holding operating system programs copyrightable: "The copyright status of the written rules for a game or a system for the operation of a machine is unaffected by the fact that those rules direct the actions of those who play the game or carry out the process." *Apple Computer, Inc. v. Formula Int'l, Inc.*, 725 F.2d 521, 524 (9th Cir. 1984) (quoting CONTU Report

at 21). "That the words of a program are used ultimately in the implementation of a process should in no way affect their copyrightability." CONTU Report at 21.

In framing its definition, CONTU emphasized that expression is copyrightable, while ideas are not: "Section 102(b) is intended, among other things, to make clear that ***the expression adopted by the programmer is the copyrightable element in a computer program***, and that the actual processes or methods embodied in the program are not within the scope of the copyright law." CONTU Report at 19 (quoting H.R. Rep. No. 94-1476, at 57 (1976); S. Rep. No. 94-473, at 54 (1975)) (emphasis in CONTU Report). "Section 102(b) in no way enlarges or contracts the scope of copyright protection under the present law. Its purpose is to restate . . . that the basic dichotomy between expression and idea remains unchanged." *Feist Publ'ns, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 356 (1991) (quoting H.R. Rep. No. 94-1476 at 57; S. Rep. No. 94-473at 54). *Lotus*'s methods of operation, which could encompass most any program, upset this delicate statutory balance.

Lotus's categorical approach also cannot be squared with the holding in *Johnson Controls* that the Court must inquire as to whether "the component in question qualifies as an expression of an idea, or an idea itself." *Johnson Controls*, 886 F.2d at 1175. Here, both the Java class libraries and their associated API specifications are copyrightable expression.

Looking in isolation at the Java class libraries, it is indisputable that they qualify for copyright protection because they are computer programs written in source code. "Source and object code, the literal components of a program, are consistently held protected by a copyright on the program." *Id.* That is true regardless of the purpose to which that program is put.

Similarly, the selection, arrangement and structure of the Java class libraries is protectable under *Johnson Controls*. *See id.* at 1174. They follow the detailed hierarchical structure described in the API specifications—the same complex structure that was presented to the Court at the March 28 hearing. *Johnson Controls* upheld the lower court's finding of copyrightability of a program structure based on far more modest evidence: that "some discretion and opportunity for creativity exist in the structure." *Id.* at 1176.

The API specifications themselves also clearly qualify for copyright protection. As the Court held, "the API specifications [] are not 'methods of operation' under section 17 U.S.C. § 102(b)." (ECF No. 433 at 11.) "API specifications are written documentation. Even if Google could show that APIs are methods of operation, that would not mean that a written work that describes or embodies APIs is automatically exempt from copyright protection." (*Id.* at 10-11.) When sufficiently expressive and original, as is the case here, written documentation qualifies for copyright protection. (*See, e.g.*, ECF No. 833 at 3 (citing cases).)

Google copied Oracle's API specifications nearly verbatim into the Android API specifications. And it created a derivative work when it implemented Oracle's detailed selection, arrangement, and structure, fixed in both the API specifications and the Java class libraries, into the source code for the Android class libraries. This is copyright infringement.

It cannot be that, when the copyrightable Java API specifications and class libraries are viewed together, they suddenly lose their copyright protection by being labeled a "method of operation" or a "system." Courts in at least three other circuits disagree with *Lotus*'s approach, and hold that identifying something as a system or method under § 102(b) does not end the inquiry into copyrightable expression. The Eighth Circuit examined the history of section 102(b) and concluded that the expression of a "system" can qualify for copyright protection:

> In light of this history of § 102(b), ***the district court's literal application of the section's language-that appellant's parts numbering system is not copyrightable because it is a "system"-cannot stand.*** All that the idea/expression dichotomy embodied in § 102(b) means in the parts numbering system context is that appellant could not copyright the idea of using numbers to designate replacement parts. Section 102(b) does not answer the question of whether appellant's ***particular*** expression of that idea is copyrightable.

Toro, 787 F.2d at 1212 (emphasis added).

Similarly, the Tenth Circuit expressly disagreed with *Lotus* and found: "We conclude that although an element of a work may be characterized as a method of operation, that element may nevertheless contain expression that is eligible for copyright protection." *Mitel, Inc. v. Iqtel, Inc.*, 124 F.3d 1366, 1370, 1372, 1373-74 (10th Cir. 1997) (reversing district court holding that command codes were method of operation but affirming holding that codes were not original).

In addition, at least one Southern District of New York case has stated it disagrees with *Lotus*: "[w]hile the Second Circuit does not appear to have decided the question, we believe that a method of operation may contain expression that it *[sic]* copyrightable." *eScholar, LLC v. Otis Educ. Sys.*, 76 U.S.P.Q. 2D (BNA) 1880, 1897 (S.D.N.Y. 2005).

A leading treatise on copyright agrees:

> The House Report makes clear that the above provision "in no way enlarges or contracts the scope of copyright protection under the present [1909] law. Its purpose is to restate, in the context of the new single Federal system of copyright, that the basic dichotomy between expression and idea remains unchanged." It would, then, be a misreading of Section 102(b) to interpret it to deny copyright protection to "the expression" of a work, even if that work happens to consist of an "idea, procedure, process, etc."

1-2 *Nimmer on Copyright* § 2.03[D] (internal citations omitted). *See also Gen. Universal Sys., Inc. v. Lee*, 379 F.3d 131, 142 (5th Cir. 2004) (stating "menu structures" are copyrightable).

The Java APIs also do not fit the *Lotus* definition of "method of operation" because they are not "a means by which a person operates something." The *Lotus* court viewed the Lotus 1-2-3 menu command hierarchy as a "method of operation" because the commands were the ***actual keystrokes*** that a person would type to use the Lotus 1-2-3 program. *Lotus*, 49 F.3d at 809; *see id.* at 815 ("The Lotus menu command hierarchy provides the means by which users control and operate Lotus 1-2-3. If users wish to copy material, for example, they use the 'Copy' command. If users wish to print material, they use the 'Print' command.").

That is not what we have in this case. On a computer or Android phone, there is no button, or key on a keyboard, or touchscreen menu labeled "HandshakeCompletedEvent," or "getCipherSuite()," or anything corresponding to the thousands of other Java API elements. If the class and method names defined in the API specification and implementation are viewed as a means "to tell the computer what to do," that is true of ***any*** part of software. All source code instructs a processor what functions to perform. Under Google's proposed interpretation of *Lotus*, nearly all computer programs would be uncopyrightable.

The Java APIs are also factually distinguishable because they are significantly more complex and expressive than the Lotus 1-2-3 menu command hierarchy. The menu structure in *Lotus* was a comparatively simple consumer user interface. Even the dissenting opinion found

"[t]he present case is an unattractive one for copyright protection of the menu. The menu commands (*e.g.*, "print," "quit") are largely for standard procedures that Lotus did not invent and are common words that Lotus cannot monopolize." 49 F.3d at 821.

III. GOOGLE'S "COMPATIBILITY" CLAIM DOES NOT JUSTIFY COPYING EXPRESSION

Google persists in mischaracterizing the Ninth Circuit's holdings in *Sega* and *Sony*. Neither case sheds any light on the issues the Court is facing here. Oracle discussed the *Sega* case in its March 23 brief. (*See* ECF No. 824 at 4-5.) Like *Sega*, *Sony* is a reverse-engineering case that concerned intermediate copying only, where the final product was not alleged to infringe. *See Sony Computer Entm't, Inc. v. Connectix*, 203 F.3d 596, 604 n.7 (9th Cir. 2000). The *Sony* court described the central premise of its holding by referencing *Sega*:

> Central to our decision today is the rule set forth in *Sega*:
>
> > Where disassembly is the *only way to gain access to the ideas and functional elements embodied in a copyrighted computer program* and where there is a legitimate reason for seeking such access, disassembly is a fair use of the copyrighted work, as a matter of law.

Sony, 203 F.3d at 602-03 (quoting *Sega Enters. Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1527-28 (9th Cir. 1993) (emphasis added in *Sony*).) Our case is not one in which copying was required to access any ideas or functional elements: Oracle publishes the API documentation for anyone to read, subject to copyright protection.

More on point is *Atari v. Nintendo*, where, unlike in *Sony* and *Sega*, the final product ***was*** alleged to infringe. 975 F.2d at 843. Applying Ninth Circuit law, the Federal Circuit found the key to Nintendo's console was entitled to copyright protection because of the "unique and creative arrangement of instructions in [its] program" and because "[e]xternal factors did not dictate the design." *Id.* at 840. While Atari could reverse engineer Nintendo's code to learn its unprotected features, it could not replicate this protected expression:

> Fair use to discern a work's ideas, however, does not justify extensive efforts to profit from replicating protected expression. Subparagraphs 1 and 4 of section 107 clarify that the fair use in intermediate copying does not extend to commercial exploitation of protected expression. *Sony Corp.*, 464 U.S. at 451, 104 S. Ct. at 793. The fair use reproductions of a computer program must not exceed what is necessary to understand the unprotected elements of the work.

Id. at 843.

Another key fact distinguishing this case from *Sony* and *Sega* is that copying in those cases was literally required for the software programs to function with the video game consoles. *See, e.g., Sony*, 203 F.3d at 599. There was no such requirement here. Google's expert admits it would have been "technically possible" for Google to write its own APIs for the 37 packages at issue. (Astrachan Dep. 251:21-252:2, 253:2-4.) Google wrote most of its own APIs and did not always take the entirety of the 37 APIs it did copy, creating an incompatible product and fragmenting Java. Google copied because it wanted to exploit the developer community Sun had cultivated over more than a decade. It copied for a commercial purpose, not compatibility.

IV. GOOGLE'S CLAIM THAT THE SUN-APACHE DISPUTE WAS JUST ABOUT BRANDING IS FALSE

In its March 27 brief, Google claimed that "[t]he dispute between Apache was about branding, and the ability to say that Apache Harmony is Java compatible." (ECF No. 831 at 3). Google is not being candid with the Court.

In 2005, Apache started the Apache Harmony project to create an open-source implementation of Java SE. The specifications for Java SE have always been made available under an openly-published license (the "specification license") that grants

> a perpetual, non-exclusive, worldwide, fully paid-up, royalty free, limited license (without the right to sublicense) ***under any applicable copyrights or patent rights it may have in the Specification*** to create and/or distribute an Independent Implementation of the Specification that: (i) fully implements the Spec(s) including all its required interfaces and functionality; (ii) does not modify, subset, superset or otherwise extend the Licensor Name Space, or include any public or protected packages, classes, Java interfaces, fields or methods within the Licensor Name Space other than those required/authorized by the Specification or Specifications being implemented; and (iii) passes the TCK (including satisfying the requirements of the applicable TCK Users Guide) for such Specification. ***The foregoing license is expressly conditioned on your not acting outside its scope.***

Java 2 Platform SE Development Kit 5.0 Specification License, *available at*: http://docs.oracle.com/javase/1.5.0/docs/relnotes/license.html. As discussed below (*see infra*, Section VI), Google has never pointed to any other license under which Apache or Google purportedly was authorized to use the Java SE specifications.

On April 10, 2007, Apache sent an open letter to Sun about its dissatisfaction with Sun's

license terms for the Java SE compatibility test kit, without which Apache Harmony would not be licensed under Sun's specification license:

> Since August 2006, the ASF has been attempting to secure an acceptable license from Sun for the test kit for Java SE. This test kit, called the "Java Compatibility Kit" or "JCK", is needed by the Apache Harmony project to demonstrate its compatibility with the Java SE specification, as required by Sun's specification license.

Open Letter to Sun Microsystems, *available at* http://www.apache.org/jcp/sunopenletter.html. Nothing in Apache's letter described the dispute as relating to access to the "Java" brand or trademarks. Consistent with the specification license, Apache acknowledged on its website that passing the JCK, also known as the TCK, was necessary not just to demonstrate compatibility, but to obtain required intellectual property rights:

> Q: Why is the TCK useful?
>
> A: It allows independent implementations to demonstrate that they are compatible with the specification, and as a result, receive all the "necessary IP" from expert group members.

FAQ, *available at* http://www.apache.org/jcp/sunopenletterfaq.html. Sun's TCK license included a field-of-use restriction that would have banned running the Apache Harmony code on mobile devices without a separate license from Sun. Apache argued imposing such a restriction was improper and refused to accept the TCK license. Sun disagreed and refused to relax its requirements; Apache never obtained a TCK license; and Apache accordingly never had a license —whether under Sun's copyrights, patents or trademarks—to implement the Java APIs in the manner it did. Apache shut down the Harmony project on November 15, 2011.

Google, along with Sun and Apache, also sat on the Executive Committee of the Java Community Process ("JCP"), where the issue was hotly debated.[2] Google engineer Bob Lee, one of Google's JCP representatives and a participant in those debates, informed Google's then-CEO

[2] *See e.g.*, Exec. Comm. Mtg. Minutes for 13-14 January, 2009, *available at*: http://jcp.org/aboutJava/communityprocess/summaries/2009/January09-public-minutes.html ("The ECs then went into private session to discuss the status of the Sun-Apache negotiations. During this session members had a lively discussion, and made some proposals that will be brought to the attention of Sun's management."); Exec. Comm. Mtg. Minutes for 24 June, 2008, *available at*: http://jcp.org/aboutJava/communityprocess/summaries/2008/June-08.summary.html ("Sun and Apache have failed to reach agreement on the terms under which the JCK would be licensed to Apache for use in the Harmony project.").

Eric Schmidt in May 2008 that ***Sun prohibited using Apache Harmony code in Android***:

> Sun puts field-of-use restrictions in the Java SE TCK licenses which prohibit Java SE implementations from running on anything but a desktop or server. These restrictions prevent Apache Harmony from independently implementing Java SE (Harmony can't put those restrictions on their own users and still Apache license the code) not to mention Android (though that's water under the bridge at this point).

(GOOGLE-27-00002479.) Schmidt responded, "I'm not surprised by Sun's position." (*Id.*) Nothing in Sun's position or analysis turned on use of the Java brand.

V. GOOGLE CANNOT ESTABLISH THE RELIANCE ELEMENT OF ITS EQUITABLE ESTOPPEL AND IMPLIED LICENSE DEFENSES

Equitable estoppel and implied license both require reasonable reliance. *See A.C. Aukerman Co. v. R. L. Chaides Constr. Co.*, 960 F.2d 1020, 1028 (Fed. Cir. 1992) (en banc) (equitable estoppel); *Effects Assocs. v. Cohen*, 908 F.2d 555, 558 (9th Cir. 1990) (implied license). The Court emphasized Google's need to establish reliance at the pre-trial conference. (*See* 3/28/2012 Hr'g Tr. 92:2-6; 95:18-96:22.) Google cannot prove reliance.

Based on Google's March 27 briefing, it is apparent Google's defenses turn almost entirely on statements made by former Sun CEO Jonathan Schwartz in a 2011 deposition, rather than on any Sun statement from 2006 when Google decided to copy from Java into Android. Google has never identified any person who supposedly relied on Schwartz's comments or any other Sun statements when making that decision. Indeed, Google's counsel stated that Google is still not "certain" if anyone is going to testify as to reliance. (3/28/2012 Hr'g Tr. 92:2-5.)

Google is precluded from putting on any such testimony at trial. Oracle served interrogatories that were squarely directed at the factual and legal bases for Google's equitable estoppel and implied license defenses. Oracle twice sought relief from the Court for Google's inadequate responses. (ECF Nos. 77, 98.) As a result, Google represented that it would make its "best efforts in good faith" to supplement its interrogatory responses. (4/6/2011 Hr. Tr. 7:25-9:22.) The Court warned the parties at the time to make a full disclosure:

> I won't make a ruling on this now, but if I was in your position I would have an exceedingly good answer, because I may encourage somebody to bring a motion to knock out a defense or knock out a claim for insufficient basis. So if you want to protect yourself against that, you should err on the side of more disclosure, and more answer, and not hide the ball. (*Id.* at 7:14-20.)

Google last supplementation was April 25, 2011 and it did not include any facts showing Google's reliance. Therefore, pursuant to FRCP 37(c)(1), Oracle will move the Court to preclude Google from presenting at trial any evidence of reliance, and to strike these affirmative defenses. *See Goodman v. Staples the Office Superstore, LLC*, 644 F.3d 817, 827 (9th Cir. 2011).

VI. GOOGLE HAS NO LICENSE DEFENSE

Google has no license defense under the GPL or Oracle's specification license. Google is not licensed under the GPL, because the Android Java API reimplementation is not also GPL-licensed (a requirement of the GPL). Google admits that it is not licensed under Oracle's specification license, and cannot qualify for it. Under the specification license, as noted above (*see supra*, Section IV), an implementation cannot superset or subset the API packages defined in the given Java API specifications, which Google admitted at the March 28 hearing it did. (3/28/12 Hr'g Tr. at 94:1-95:6.) Moreover, as Google acknowledged in its answer and counterclaim: "The *only* way to demonstrate compatibility with the Java specification is by meeting all of the requirements of Sun's Technology Compatibility Kit ('TCK') for a particular edition of Sun's Java." (ECF No. 32 at 12-13 (emphasis added).) Google has not tested Android with the TCK.

Oracle's interrogatory 15 sought the factual and legal bases for Google's license defense. Google's last supplemental response did not identify a single license granted by Sun or Oracle that applies to the 37 APIs at issue. And indeed, when questioned by the Court at the recent hearing, Google did not identify any license other than the specification license, which it agreed did not apply to it. (3/28/2012 Hr'g Tr. 95:18-96:22.)

VII. GOOGLE'S JAVA API DOCUMENTATION IS SUBSTANTIALLY SIMILAR TO ORACLE'S CORRESPONDING DOCUMENTATION

Google does not dispute that "the Android specifications for the 37 API packages at issue have substantially the same selection, arrangement and structure of API elements as the J2SE specifications." (ECF No. 778 at 3.) Google's documentation reproduces the API elements from Oracle's documentation, which is enough to establish infringement. Google argues, however, that the English prose descriptions of the elements in the two specifications are sufficiently

different to avoid infringement. Google is incorrect. Google's plagiarism and paraphrasing of Oracle's English-language descriptions also constitutes infringement. Just as Google copied the selection, coordination, and arrangement of the 37 Java APIs into the Android class library code, it copied the APIs into its documentation. Google's descriptions show that the Android APIs follow the same detailed plot line as the Java APIs. Copyright "cannot be limited literally to the text, else a plagiarist would escape by immaterial variations." *Sid & Marty Krofft Television Prods., Inc. v. McDonald's Corp.*, 562 F.2d 1157, 1167 (9th Cir. 1977) (quoting *Nichols v. Universal Pictures Corp.*, 45 F.2d 119, 121 (2d Cir. 1930) (Hand, J.)). Other courts have similarly found infringement in cases of paraphrasing:

> The language of LSI's modified MPO program is likewise similar to that of K-T's materials: their ***paragraphs are about the same size***, their ***phrases are similar***, their ***ideas are presented in the same order***; in short, parts of the modified MPO program are but a transparent, syntactic rearrangement of portions of K-T's copyrighted materials. ***While no longer identical to those materials, the modified MPO program still bears many telltale signs of its origins. It is still a copy--still a child of infringement***.

Kepner-Tregoe, Inc. v. Leadership Software, Inc., 12 F.3d 527, 536 (5th Cir. 1994) (emphasis added) (rejecting defendant's "surgical efforts to remove only the infringing language").

The Android documentation itself belies Google's attempts to downplay its similarity to the Java documentation. Space constraints prevent Oracle from providing more than this example of how Google plagiarized its descriptions:

Oracle's J2SE 5.0 Specification for javax.crypto.CipherInputStream[3] (excerpts)	**Google's Android Specification** for javax.crypto.CipherInputStream[4] (excerpts)
The Cipher must be fully initialized before being used by a CipherInputStream. For example, if the Cipher is initialized for decryption, the CipherInputStream will attempt to read in data and decrypt them, before returning the decrypted data.	The cipher must be initialized for the requested operation before being used by a CipherInputStream. For example, if a cipher initialized for decryption is used with a CipherInputStream, the CipherInputStream tries to read the data an decrypt them before returning.

[3] http://docs.oracle.com/javase/1.5.0/docs/api/javax/crypto/CipherInputStream.html
[4] http://developer.android.com/reference/javax/crypto/CipherInputStream.html

Dated: April 3, 2012

MORRISON & FOERSTER LLP

By: /s/ Michael A. Jacobs

Attorneys for Plaintiff
ORACLE AMERICA, INC.